

OHDFewsadapter

Background

OHDFewsadapter is a java class used by all OHD modules in CHPS. It is the initial point of interaction between all OHD modules and FEWS. FEWS is configured to use the OHDFewsadapter through FEWS workflows.

Below is part of a workflow used to run SNOW-17 in FEWS.

```
<?xml version="1.0" encoding="UTF-8"?>
.....
<activities>
.....
  <executeActivities>
    <executeActivity>
      <description/>
      <command>
        <className>ohd.hseb.ohdfewsadapter.OHDFewsAdapter</className>
      </command>
      <arguments>
        <argument>run_info_file = %ROOT_DIR%/run_info.xml</argument>
      </arguments>
      <timeOut>10000</timeOut>
    </executeActivity>
  </executeActivities>
.....
</activities>
</generalAdapterRun>
```

The <executeActivity> workflow process (highlighted in blue above) executes the OHDFewsAdapter.main() java method with 1 argument.

OHDFewsAdapter Argument

OHDFewsAdapter's argument is specified in a FEWS workflow using the "property = value" construct. The only required property is an xml file with model run information (run info file). This argument is a fully qualified path and filename to the run info file.

OHDFewsAdapter Run Info XML File

The OHDFewsadapter run info xml file contains information needed to run all OHD models. An excerpt is shown below.

```
<timeZone>0.0</timeZone>
<startDateTime date="2007-10-22" time="12:00:00"/>
<endDateTime date="2007-10-31" time="12:00:00"/>
<time0 date="2007-10-25" time="12:00:00"/>
<lastObservationDateTime date="2007-10-25" time="12:00:00"/>
<workDir>/awips/CHPS0810/home/grijzea/ohdrfc_sa/Modules/ohd /lagk/work</workDir>
<inputParameterFile>/awips/CHPS0810/home/grijzea/ohdrfc_sa/Modules/ohd /lagk/input/params.xml</inputParameterFile>
<inputStateDescriptionFile>="/awips/CHPS0810/home/grijzea/ohdrfc_sa/Modules/ohd /lagk/work/states.xml</inputStateDescriptionFile>
<inputTimeSeriesFile>/awips/CHPS0810/home/grijzea/ohdrfc_sa/Modules/ohd /lagk/input/inputs.xml</inputTimeSeriesFile>
<outputTimeSeriesFile>/awips/CHPS0810/home/grijzea/ohdrfc_sa/Modules/ohd /lagk/output/outputs.xml</outputTimeSeriesFile>
<outputDiagnosticFile>/awips/CHPS0810/home/grijzea/ohdrfc_sa/Modules/ohd /lagk/output/log.xml</outputDiagnosticFile>

<properties>
  <string key="legacyLocation" value="/awips/CHPS0810/home/grijzea/ohdrfc_sa/Modules/ohd/bin"/>
  <string key="outputLocationId" value="SELWE"/>
```

```
<int key="printDebugInfo" value="0"/>
</properties>
```

The file is separated into two parts. The first part consists of at least 6 required elements (highlighted in yellow). Note: Although `inputParameterFile`, `inputStateDescriptionFile`, `inputTimeSeriesFile`, and `outputTimeSeriesFile` are not required by the xml schema, all OHD models require these elements¹. The second part is a `properties` element with 1 or more sub-elements and properties defined.

The `OHDFewsadpater` `properties` element has 1 required property and several more optional ones.

Optional Properties

- `printDebugInfo` (valid values are 0 to 5; the default is 0)

The `printDebugInfo` property is used to set the level for debug output. A higher number represents more debug output.

When the default is used (i.e. “0”), only ERROR, WARNING, and INFO messages are written for OHD models. This logging is included in the same log file FEWS uses (i.e. `log.txt`, and also appears in the GUI’s bottom panel).

When `printDebugInfo` is > 0 , additional logging is written to an xml file specific to the OHD model. This file is included in the “dump” of files generated when running an OHD model in debug mode. The file is located in the `output/` directory and is named `diag.xml`.

- `outputLocationId` (a string identifying the output location)

The `outputLocationId` property is used to explicitly identify the `locationId` for model output time series. When not defined the location id for one of the inputs is used for all output time series.

NOTE: for models whose input is from an upstream basin (e.g. channel routing models), the `outputLocationId` should most likely be set.

- `legacyLocation`: a directory name

The legacy location is used to set the location for the binary executable for OHD models not ported to Java

- `startLocalHourOfDay`: an integer to set the starting local hour (0-24) of the day
- `startHourOfDay`: an integer to set the starting hour (0-24 in GMT) of the day - *Default is 12*

¹ The one exception is the BASEFLOW model, which in some cases does not need an `inputTimeSeries` element.

NOTE: OHDFewsadapter calculations are done internally in GMT and by default the starting hour of the day is assumed to be 12GMT. In some cases (ESP and Calibration) this assumption is not valid.

The variables *startLocalHourOfDay* and *startHourOfDay* have the same purpose (either can be used).

If "startLocalHourOfDay" is used it is combined with the <timeZone> setting at the top of the GA config to identify the time in GMT

If "startHourOfDay" is used the hour is expressed as a GMT hour for the start of the day, regardless of what the <timeZone> setting is set to.

Input Time Series Mods

A few OHD models (sac, sacht, apicont, snow, ssarresv, and ressnl) are capable of receiving mods as separate input time series. In these cases 2 or 3 <inputTimeSeriesFile> elements will appear in the runInfo.xml file. Below is an example for SNOW17, which can have up to 3 elements (one for input time series, one for equidistant mod time series, and one for non-equidistant mod time series).

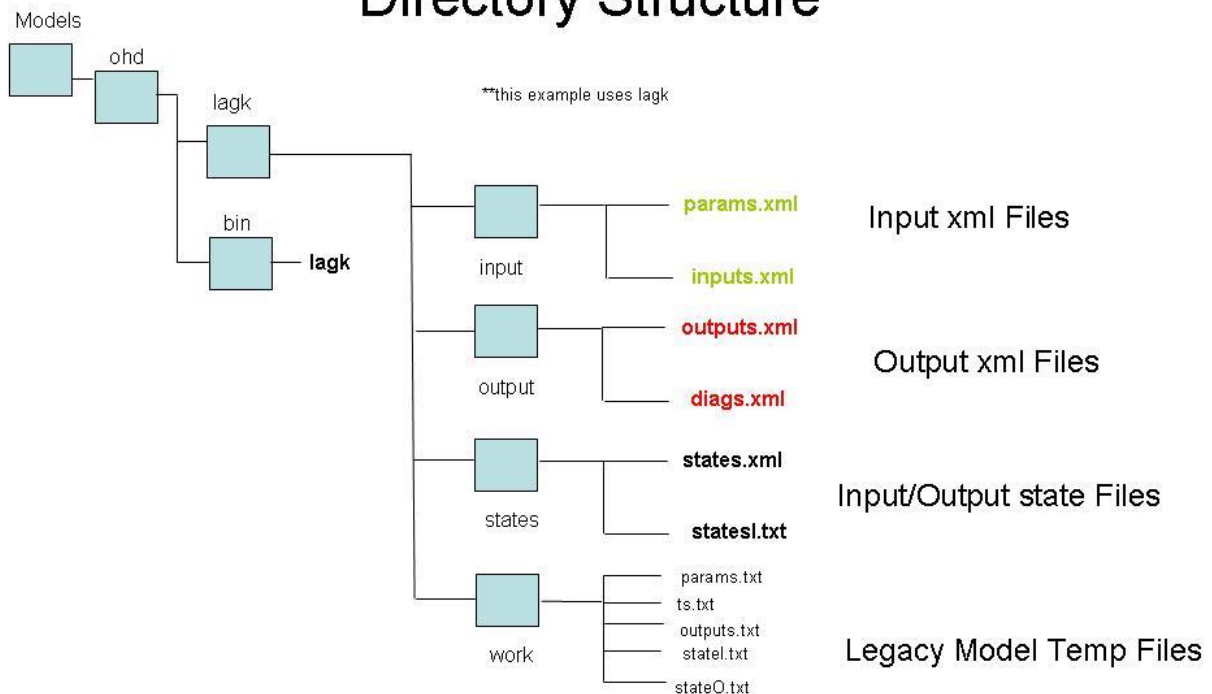
```
<inputParameterFile>snow17/forecast_isac1/input/params.xml</inputParameterFile>
<inputStateDescriptionFile>snow17/forecast_isac1/work/states.xml</inputStateDescriptionFile>
<inputTimeSeriesFile>snow17/forecast_isac1/input/inputs.xml</inputTimeSeriesFile>
<inputTimeSeriesFile>snow17/forecast_isac1/input/wechng_mods.xml</inputTimeSeriesFile>
<outputDiagnosticFile>snow17/forecast_isac1/output/diag.xml</outputDiagnosticFile>
```

Model Directory Structure

The following diagram shows the recommended directory structure for CHPS models in FEWS. Some things to note are:

1. The legacy executable, in this case 'lagk' is under ./Models/ohd/bin
2. For legacy models ported to Java, the ./Model/ohd/bin directory is not needed

Recommended OHDFEWSAdapter Model Directory Structure



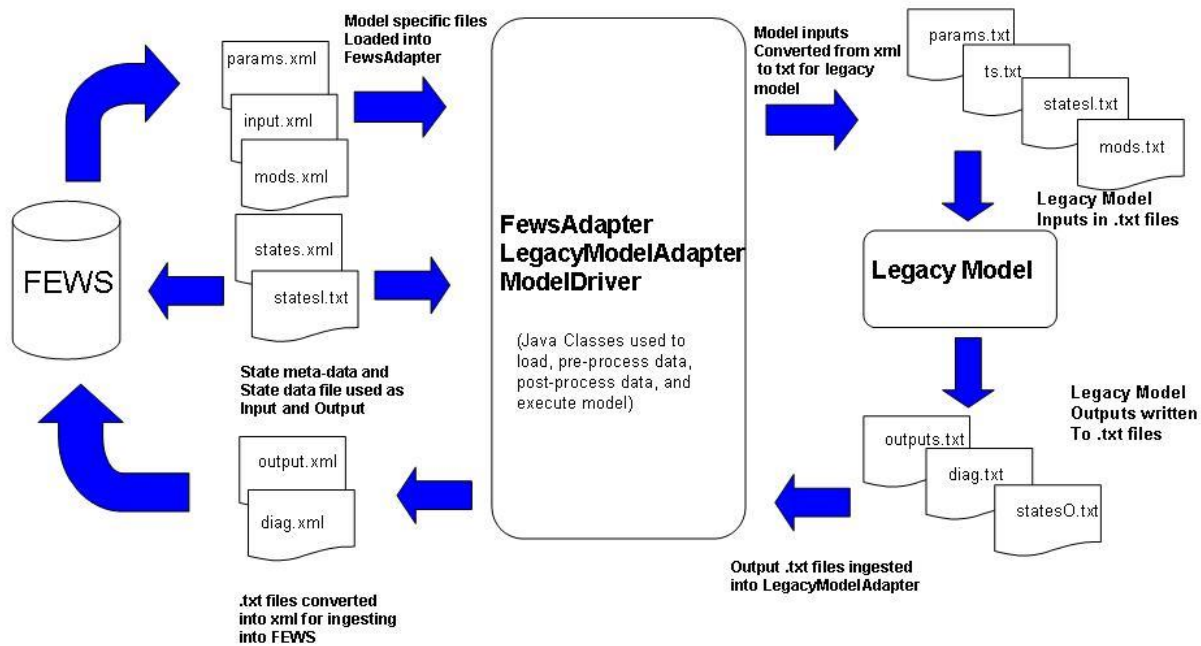
How it Works

The OHDFewsAdapter java class is the initial point of interaction between FEWS and the OHD models (C/FORTRAN and Java based). In addition to **instantiating the appropriate model driver**, OHDFewsAdapter is where the processes to **parse and load the input time series, parameters, and states** into the appropriate objects are initiated. After the data is loaded, OHDFewsAdapter **calls the model driver's execute method**.

In the case of non-java legacy models the model driver's execute method is responsible for instantiating a LegacyModelAdapter which creates input time series flat files from objects stored in the model driver class, invokes the legacy model, and converts output flat file time series data into an output time series data object. Finally OHDFewsAdapter takes the data object returned by the legacy adapter and initiates the process used to **convert the output data object into an output FEWS xml time series file**.

Data Flow

NWSRFS Legacy Model in CHPS



Data Flow

The figure above shows the flow of data between FEWS and OHDFewsadapter.

Input Data

Input data as xml files (`params.xml`, `inputs.xml`, `states.xml`, `run_info.xml`) and 1 text file (initial state values) are exported from FEWS and passed to the OHDFewsadapter for each model. For Java models, the xml files are used directly, while for non-Java models the xml files are converted to native-format text files.

Output Data

Output data as xml files (`outputs.xml`, `states.xml` and `diag.xml`) and 1 text file (output states) are passed from the OHDFewsadapter back to FEWS. For non-Java models output text files (`output.txt` and `diag.txt`) are converted to xml files before sending back to FEWS.

FEWS Configuration files

OHD modules use FEWS' General Adapter (GA-put in a link). The GA module configuration provides unique information (module name, module data, module properties, etc.) in the common format described above. Sample GA module configurations for each OHD module are provided as part of the module's documentation.

Each OHD module executed through the General Adapter uses the same basic FEWS configuration files.

Configuration File	Description
Module (GA) Forecast UpdateStates	<p>Executes the OHD module by first identifying all input, parameters, states (optional), and properties/run time options needed by the module and after the module has completed identifying the data produce by the module for ingesting into FEWS.</p> <p>Most OHD modules have 2 versions of the GA module (Forecast and UpdateStates). The UpdateStates is unique in that it defines a looping interval for subdividing the run period into successive shorter runs. At the end of each short run the module's states are ingested into FEWS.</p>
Module Parameters	A moduleParameter.xsd file that uses basic types of data (string, double, int, table) to hold model specific parameters in xml tags.
Parameters (Data Types)	Used to identify the input/output Module data types (e.g. SQIN, QINE). Note: One file is used for all FEWS modules. If the data type already exists, there is no need to add it again.
Workflows Forecast UpdateStates	<p>Controls and defines the order in which to execute a series of modules. Other features like looping, whether or not a module can be execute of a previous modules success/failure is also configured in this file.</p> <p>There is a separate UpdateStates and Forecast workflow for executing the UpdateStates and Forecast modules respectively.</p>
Module Instance Descriptors	Used to define each module (e.g. module name, type). Type refers to whether the module is a FEWSTransformation or GeneralAdapter. All OHD modules are of type GeneralAdapter.
Workflow Descriptors	Used to define each Workflow's characteristics. Each OHD module does not need a workflow, but

	rather executes as part of a parent workflow (e.g. OFS Segment). There are various options for defining a workflow that are defined by looking at the schema.
Id Mappings (optional) Import Export	Used to map Input/Output time series characteristics (id, parameter, units, etc.). This is important for ensuring the time series characteristics are defined in a way both FEWS and the OHDFewsadapter recognize. There is a separate file for configuring imports (to FEWS) and exports (to OHDFewsadapter).
Locations	Used to identify the geographic location of data (basin/point). Location (locatoId) is a basic property of all-time series. Note: for Grid data, “locationId” is a surrogate for identifying projection information (defined in Grids.xml).

Extra Details

Displays

Another piece of configuring each module are displays. Display configurations are not needed for executing the OHD modules in FEWS. These configurations are useful for viewing module output. Some OHD modules have unique displays, others reuse common displays. Below is a list of some FEWS display configuration files. More detail is available in the FEWS documentation.

1. [DisplayGroups.xml](#)
2. [TimeSeriesDisplayConfig.xml](#)
3. [SpatialDisplay.xml](#)

Creating Groupings (i.e. Sets)

FEWS allows the grouping of modules and locations. This becomes useful when an action (FEWS Transformation, Display definition, Export/Import of time series) is desired for more than one module/location. Using “Sets” avoids the need to create repetitive entries in a FEWS configuration file. The First step is to create module sets and location sets using FEWS’ [moduleInstanceSets.xml](#) and [locationSets.xml](#) files.